COBB

July 1992
Vol. 3 No. 7

INSIDE DOS™

Tips & techniques for MS-DOS & PC-DOS

Understanding your computer's memory

Most DOS users don't think much about the random access memory (RAM) in their personal computers until they upgrade either their system or their software. For example, if you're considering buying a more powerful machine, you'll notice that you pay several hundred dollars more for a few extra megabytes of RAM. You might justifiably wonder if the additional RAM is worth the expense.

A more jarring way of confronting terms like RAM and memory is through an application's error message. You might install new software on your system only to see a message like *Not enough memory to run application* when you try to start the new program.

As you can see from these examples, RAM can be costly, yet it's vital to some applications. But why? In this article, we'll show you why RAM is important and try to answer some other common questions about the memory in your system. Let's begin by looking at some basic RAM characteristics.

What is RAM?

As we mentioned earlier, RAM stands for "random access memory," but those three words don't seem to mean much more than the acronym. Let's take a closer look at some important aspects of RAM. The RAM in your computer

- allows your system to read and write information
- allows programs to access information randomly
- holds the instructions of programs you run
- lets you make changes without writing to the disk
- loses information when you turn off your PC

Reading and writing

Perhaps the most basic aspect of RAM is that DOS and other programs can both read information from and write information to RAM. As you may know, this differs from read-only memory (ROM), which stores the basic information your computer needs to run. ROM is perfect for storing the fundamental instructions you need for booting up but, as we'll see, RAM becomes more important when you run applications.

Random access

As you might guess, "random access" is another important aspect of RAM: DOS can get information from this type of memory randomly. In other words, DOS doesn't have to read each character in RAM in order to find the information it's looking for. (Sorting through all the information to find what you're looking for is called *sequential access*.) This ability lets RAM read data very quickly. For example, imagine if you had to locate a name in a phone book sequentially to find the phone number for Clyde Zellers. You'd have to begin with the first entry under A and scan hundreds of thousands of entries until you reached Zellers. But, intuitively, you'd know to use "random access" to look up this telephone number: You'd turn to the back of the book to find the Z listings, then you'd refine your search to the entries beginning with Ze, until you finally located Clyde Zellers' number. RAM also is efficient at locating information. However, instead of using the alphabet to create an index, RAM indexes programs and other files by their address, or location in memory.

Continued on page 10

IN THIS ISSUE

A Publication of The Cobb Group

INSIDE DOS™

Inside DOS (ISSN 1049-5320) is published monthly by The Cobb Group.

Prices
Domestic .... $59/yr. ($7.00 each)
Outside the US .... $79/yr. ($8.50 each)

Address
The Cobb Group
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Phone
Toll free .... (800) 223-8720
Local .... (502) 491-1900
FAX .... (502) 491-8050

Staff

Editor-in-Chief	Suzanne Thornberry
Consulting Editors	Jim Welp David Reid Tim Landgrave
Contributing Editor	Van Wolverton
Editing	Clyde Zellers Polly Blakemore Cecilia Crosby-Lampkin
Production	Debbie Lane
Design	Karl Feige
Publications Manager	Elayne Noltemeyer
Publisher	Douglas Cobb

Address correspondence and special requests to The Editor, *Inside DOS*, 9420 Bunsen Parkway, Suite 300, Louisville, KY 40220. Address subscriptions, fulfillment questions, and requests for bulk orders to Customer Relations, at the same address.

Postmaster: Send address changes to *Inside DOS*, P.O. Box 35160, Louisville, KY 40232. Second class postage is paid in Louisville, KY.

Conventions

To avoid confusion, we would like to explain a few of the conventions used in *Inside DOS*.

When we instruct you to type something, those characters usually appear on a separate line along with the DOS prompt. The characters you type will appear in red, while the characters DOS displays will appear in black.

Occasionally, we won't display the command you type on a separate line. In these cases, we'll display the characters you type in italics. For example, we might say, "Issue the command *dir *.txt* at the DOS prompt." Although DOS is not case-sensitive, we'll always display the characters you type in lowercase.

When we refer to a general DOS command (not the command you actually type at the DOS prompt), we'll display that command name in all caps. For example, we might say, "You can use either the COPY or XCOPY command to transfer files from one disk to another."

Many commands accept parameters that specify a particular file, disk drive, or other option. When we show you the form of such a command, its parameters will appear in italics. For example, the form of the COPY command is

```
copy file1 file2
```

where *file1* and *file2* represent the names of the source file and the target file, respectively.

The names of keys, such as [Shift], [Ctrl], and [F1], appear in brackets. When two keys must be pressed simultaneously, those key names appear side by side, as in [Ctrl][Break] or [Ctrl]Z.

LETTERS

A workaround for finding text in large files

I attempted to use the FIND technique presented in the article "Increase Your DOS Command-line Power with Pipe Filters," which appeared in the April 1992 issue of *Inside DOS*. As you explained in the article, I replaced *FILENAME.TXT* with the name of the file I was searching and enclosed in quotation marks the text string I wanted to locate. However, when I tried to use the pipe to find a text string in a large file, I received the message *Intermediate file error during PIPE. Access denied.* Do you know of a workaround for this problem?

Peter Barron
New York, New York

When you receive this message, DOS is telling you that it can't create the large temporary files it needs to process the command. In fact, the temporary file is a copy of the file you type when you issue a command in the form

```
type FILENAME.TXT ¦ find "string"
```

Consequently, if you don't have room on your hard disk for another copy of the file, you won't be able to pipe it through the FIND command.

Fortunately, you can use the FIND filter without creating the temporary files required by piping. The trick is to use redirection to send the file directly into the FIND filter. With redirection, the command to find a text string within a file takes the following form:

```
FIND "string" < FILENAME.TXT
```

As in the piping technique, you enclose the text you want to find in quotation marks. However, in this version of the FIND command, you don't use TYPE *FILENAME.TXT*. Instead, the < redirection symbol tells DOS to send the file directly to the FIND command for processing, which saves DOS from creating a temporary file.

Using the FIND command with redirection produces the same result as using it with TYPE and piping. Returning to the example in the April 1992 article, if you enter the command

```
C:\>find "Kentucky" < company.txt
```

DOS will display the same lines of COMPANY.TXT containing the word *Kentucky*:

```
Kentucky Board of Education
Commonwealth Insurance of Kentucky
First Kentucky Savings and Loan
```

Putting directory symbols to good use

Like many DOS users, I often wondered if there were any uses for the .. symbol, which stands for the parent of the current directory, and the . symbol, which stands for the current directory. Of course, I soon discovered that *cd..* is the quickest way to change to the parent directory. But I'd also like to remind your readers of a few other uses for these directory symbols.

You can use . in any command to represent the files of the current directory. You'll save two keystrokes over typing the more familiar *.*. For example, to copy all of the files from a directory named C:\HOUSE\ROOMS to the A: drive, you'd simply enter the command

```
C:\HOUSE\ROOMS>copy . a:
```

You also can copy all of the files in the current directory to the parent directory by typing *copy*, a space, a period to represent the files in the current directory, another space, and the two periods to represent the parent directory. For example, if you typed

```
C:\HOUSE\ROOMS>copy . ..
```

DOS would copy all the files in the C:\HOUSE\ROOMS directory to the C:\HOUSE directory.

You can even use the . and .. symbols with the delete command. For example, instead of typing

```
C:\HOUSE\ROOMS>del *.*
```

to delete all of the files from the C:\HOUSE\ROOMS directory, you can type

```
C:\HOUSE\ROOMS>del .
```

As you can see, the . and .. symbols can save you time in entering several different commands. I hope other *Inside DOS* readers will give them a try.

Dave Chapman
Sacramento, California

Thanks for the tips, Dave. Almost every DOS user is on the lookout for ways to save a few keystrokes.

What is COMSPEC?

In the article "Increase Your DOS Command-line Power with Pipe Filters," which appeared in the April 1992 issue, you show the output from the SET command. I understand the PROMPT and PATH settings, but I'm not familiar with COMSPEC. What does this setting do?

Robert Kriebel
Huntington, West Virginia

Like PROMPT and PATH, COMSPEC is an *environment variable*. But unlike PATH or PROMPT, you normally don't need to set COMSPEC. Instead, DOS will record and track this variable for you. Specifically, DOS uses COMSPEC to note the location of COMMAND.COM when you boot up. This important file contains the instructions for all internal DOS commands, such as DIR, PATH, and PROMPT. (The instructions for external commands, on the other hand, are in the COM files in your DOS directory.)

Typically, COMMAND.COM is in your root directory. Therefore, if you type *set* and press [Enter], one of the lines will probably display the following information:

```
COMSPEC=C:\COMMAND.COM
```

As you can see, DOS will take care of the COMSPEC variable, so you don't have to. In fact, you could crash your system if you haphazardly switch the COMSPEC. Unless you're an advanced DOS user, you'll probably just want to let DOS handle this environment variable.

Incidentally, you may have noticed that the DOS 5 setup program copies COMMAND.COM to both the root directory and the C:\DOS directory. If you'd like, you can delete the extra copy of COMMAND.COM from your C:\DOS directory to save disk space. However, you should never delete COMMAND.COM from your root directory. If you do, DOS won't be able to boot your computer from its hard disk. Instead, it will halt and present the message *Bad or missing Command Interpreter*. (In this situation, you can use an emergency system diskette to reboot your computer, as Van Wolverton explains on page 4.)

DOS User's Journal

If you haven't upgraded to DOS 5, you're not alone. Many of our readers have told us they haven't upgraded and don't plan to in the near future. If you're running a version of DOS previous to DOS 5, we've created a new journal just for you.

In *DOS User's Journal*, we provide the same types of tips, techniques, and batch files you're used to seeing in *Inside DOS*, but ones that are designed for DOS versions 2.1-4.01. To transfer your *Inside DOS* subscription to *DOS User's Journal* free of charge, call us at (800) 223-8720 or fax us at (502) 491-8050.

VAN WOLVERTON

Protecting your data and equipment from disaster

Office of Emergency Preparedness. Office of Emergency Planning. Disaster Services. Federal Emergency Management Agency. Damage Control. When floods, tornados, earthquakes, and other natural disasters overwhelm us, these offices try to relieve the suffering and repair the damage. The better they have planned their response, the better they're able to cope with the chaos and destruction left behind.

On a much smaller scale, we have to consider the consequences of misfortune—disaster seems too strong a word—that can befall our computers. The more we rely on the computer for our livelihood the greater the cost of losing the system or the data stored in it; the greater the potential loss, the greater the need to prevent misfortune and plan how to respond should it occur.

Computer misfortunes fall into two broad categories: loss of or damage to the computer itself, and loss of data stored on the hard disk. Although the computer system represents a significant capital investment, in most instances the potential cost of lost data can easily outstrip the cost of the entire system.

This article will ignore what is probably the most common computer misfortune: the minor damage to one part of the system that defies immediate repair, usually happens at the worst possible time, and occurs under circumstances impossible to predict or prevent. Examples abound: the cat chews the mouse cord in half; your precious three-year-old daughter pours a glass of orange juice from a supposedly spill-proof glass onto the keyboard, carefully making sure that each key gets its share; for no apparent reason, a prized bust of Mozart falls from its shelf onto the printer, shattering the plastic shield and cracking the print head. You can't plan for these unimagined events; each is to be savored as its own unique experience. All you can do is pay the repair bill and wish you'd never been born.

Protecting your equipment

There's no magic to disaster prevention. In fact, you've probably heard most of the cautions, but they're worth repeating. Protecting your system is fairly simple:

- Keep the system in a room whose temperature and humidity are comfortable for you.
- Dust is the enemy of electronic equipment. Get some dust covers and use them when the system is turned off (too much heat will build up if you cover the system while it's running). Invest in one of those miniature vacuum cleaners and give the system a good cleaning every three months or so.
- Once you turn the system on, don't turn it off until you're finished with it for the day.
- Keep drinks of all kinds away from the keyboard; it's almost impossible to clean one up after a drink has been spilled on it.
- Plug your computer into a surge suppressor. If thunderstorms or power outages are common in your area, invest a few hundred dollars in an uninterruptible power supply; it will run your system for 10 or 15 minutes after a power failure, giving you time to save files and shut the system down (and unplug it if thunderstorms are causing the problem).
- Check with your insurance agent and make sure that your policy covers the replacement cost for both hardware and software.

Protecting your data

Protecting your data is even easier. Most important: Back up your hard disk regularly. It's the cheapest and most effective step you can take to reduce the pain and suffering caused by a hard disk failure. You can, of course, spend quite a bit buying backup programs and tape drives, or even removable hard disks, to expedite the backup process, but you don't have to spend a nickel if you don't want to. The BACKUP and RESTORE commands that come with DOS, while not as fast or convenient as backup programs, will nonetheless back up onto a diskette fairly quickly the data files that you have changed. If you do a little planning—for example, you can create a batch file to automate the entire backup process so that all you have to do is swap diskettes—a daily backup shouldn't take more than four or five minutes.

Version 5 of DOS includes the UNDELETE and UNFORMAT commands. You should use the MIRROR command to give them the best possible chance of recovering files that you inadvertently delete or disks that you inadvertently format. In fact, you should use MIRROR twice.

First, use the MIRROR command with the /PARTN parameter to copy the hard disk partition table to a diskette. This command is straightforward:

```
C:\>mirror /partn
```

DOS will prompt you for the drive letter that contains the diskette where the file is to be written. To keep your preparations as neat as possible, you can copy the partition table to your emergency system diskette (described a bit later) and store it in a safe place.

Second, put a MIRROR command in the AUTOEXEC.BAT file. The MIRROR command will save the latest version of the File Allocation Table and root directory of your hard disk in a file and will turn on deletion tracking, which saves directory information about deleted files. In short, MIRROR saves the information for the UNDELETE command. Here's what to put in the AUTOEXEC.BAT file:

```
loadhigh c:\dos\mirror c: /1 /tc
```

MIRROR leaves a terminate-and-stay-resident (TSR) program in memory to keep track of the files that you delete. The LOADHIGH command tells DOS 5 to put this TSR in high memory, if possible, so that it doesn't reduce the amount of memory available to your other programs. If you aren't using version 5, or your system doesn't have an 80386 or 80486 processor, DOS can't load the program into high memory, so omit *loadhigh* from the command; MIRROR will take up about 6500 bytes of memory.

Although MIRROR can give you a little insurance against accidentally deleting a file, there's only one way to protect against data lost in a disaster like the fires that swept the Oakland, California, hills in 1991: Periodically back up your hard disk and store the backup diskettes or tapes in a different location. Companies, especially financial institutions, routinely do this, storing even daily backups in ultra-safe facilities such as abandoned mines or railroad tunnels. Your situation may not warrant such measures. The deciding factor is how painful the loss of your data would be. Only you know that.

Protecting CONFIG.SYS and AUTOEXEC.BAT

The AUTOEXEC.BAT and CONFIG.SYS files in your hard disk's root directory are critical to the operation of your computer. They tell DOS how to manage the devices and memory installed in your computer. Unfortunately, many programs that you install change these files, sometimes without warning you or giving you a chance to make the changes yourself. To make sure you can recover these files, copy them to your emergency system diskettes (to be described in a moment).

There's an additional step you can take with AUTOEXEC.BAT that eliminates the possibility of other programs making changes you don't want. Change the name of AUTOEXEC.BAT to something else, such as REALAUTO.BAT. Then create a new AUTOEXEC.BAT that contains just one command:

```
@realauto
```

The @ sign tells DOS not to echo the REALAUTO command. You could omit that if you don't mind seeing the command displayed when your system starts.

Now when an installation or setup program changes AUTOEXEC.BAT, you'll have no trouble figuring out what the changes were. You can add any new commands to REALAUTO.BAT and change the values of commands like PATH, if necessary, then cut AUTOEXEC.BAT back to one command.

The emergency system diskette

If you—or a helpful installation program—make a change to AUTOEXEC.BAT or CONFIG.SYS that keeps your system from starting properly, you're stuck. You've got to remove the offending change, but you can't start the system to edit the file. The only way to start your system is to use a system diskette.

But we're approaching a Catch 22 here. Starting with version 4 of DOS, you don't get a system diskette with DOS; you must create it yourself. If you haven't created the system diskette, the installation diskettes won't do you any good unless you have access to another computer that you can use to create a system diskette.

And it may not be much help if you find an old version 3.3 system diskette, because it can't handle hard disk partitions larger than 32 Mb. If you've got a hard disk larger than 32 Mb that you formatted with DOS version 4 or 5, you'll be able to start DOS with that version 3.3 system diskette, but you won't be able to edit AUTOEXEC.BAT or CONFIG.SYS because your version of DOS can't read the hard disk (when you try to use the C: drive, DOS will present the message *Invalid drive specification*).

It takes just a few minutes to create a system diskette, so do it now and avoid these potential problems. Choose a diskette of the maximum capacity supported by your drive. For example, if you're using 5.25″ diskettes, choose a 1.2 Mb capacity diskette if your drive can handle high-density diskettes. Or, for 3.5″ diskettes, choose a 1.44 Mb capacity diskette, rather than a 720 Kb one. (The high-density diskettes in both sizes are designated DS/HD, for double-sided, high-density; the lower density diskettes are designated DS/DD, for double-sided, double-density.) It's worth using a high-density diskette for making an emergency system diskette, because you'll be able to copy more files to it.

Put the diskette (either new or one that contains files you no longer need) in drive A and type

```
C:\>format a: /s
```

The /S parameter tells DOS that after it has formatted the disk it should copy the three files needed to start the system: two hidden files named IO.SYS and MSDOS.SYS and COMMAND.COM. DOS formats the diskette then responds *Format complete* and *System transferred*.

Although this diskette will start DOS, you have room to add some utilities and commands that might be useful. If you need it to start your system, there's a chance that you're going to have to do some repair work, so you'd better put some tools on the diskette.

First, as mentioned earlier, when you use the MIRROR command to save the partition information from your hard disk, put this system diskette in drive A. Next, copy the AUTOEXEC.BAT and CONFIG.SYS files from your hard disk to the system diskette.

Even if you're using 360 Kb diskettes, there's still enough room on the system diskette for some DOS tools. We suggest that you copy the following files from \DOS on the hard disk to the system diskette:

EDLIN.EXE
DEBUG.EXE
MIRROR.COM
UNFORMAT.COM
UNDELETE.EXE
XCOPY.EXE

This gives you the minimum set of tools you'll need to fix most problems. If you're using DOS 5 and 720 Kb diskettes, copy these files, too, to add the DOS Editor and QBasic:

EDIT.COM
QBASIC.EXE

Finally, if you're using DOS 5 and 1.2 Mb or 1.44 Mb diskettes, add these files for a complete toolkit:

ATTRIB.EXE
BACKUP.EXE
CHKDSK.EXE
DOSSHELL.COM
DOSSHELL.EXE
EXPAND.EXE
FDISK.EXE
FORMAT.COM
MODE.COM
RESTORE.EXE

When your system diskette is complete, use the DISKCOPY command to make a copy of it. Keep one copy with your other diskettes near your computer, but keep the second copy separate from the others. It doesn't have to be in a bomb-proof cavern—the Cold War has ended, after all—but it wouldn't hurt to put it in a different room, or maybe the refrigerator. It takes just a few minutes to make this copy and put it somewhere else. Then, if you run into trouble, you just place the system diskette in the A: drive, close the drive door, and press [Ctrl][Alt][Del]. If your problem was caused by the hard drive, you'll soon see the A:\> prompt. Now, you can begin trying to trace down your problem.

Contributing editor Van Wolverton is the author of the best-selling books Running MS-DOS 5 *and* Supercharging MS-DOS. *Van, who has worked for IBM and Intel, currently lives in Alberton, Montana.*

BATCH FILE TIP

Removing *@echo off* helps you debug batch files

Many batch file programmers begin each batch file with the *@echo off* command. As you probably know, the *echo off* portion of this command turns off echo, thus preventing DOS from displaying the batch file's commands as it carries them out. In DOS versions 3.3 and later, you can precede *echo off* with the symbol @. The @ symbol suppresses the display of the *echo off* command itself. (In fact, in DOS versions 3.3 and later, you can suppress the display of *any* command in a batch file by preceding it with the @ symbol—not just the *echo* command.)

Obviously, beginning your batch files with *@echo off* makes for a tidier display when you run the batch file. Unfortunately, by turning off echo, you'll often mask errors in your batch files that you could otherwise quickly discover. For example, if your batch file contains a syntax error and you've turned off echo, you won't be able to tell which command contains the error. DOS will simply display the message

```
Syntax error
```

and return you to the command prompt.

When a new batch file doesn't work as you expect, you need all the help you can get to figure out what's gone wrong. If you let DOS display the commands as it carries them out, you'll see which line contains the syntax error.

Fortunately, you can easily edit the batch file so it will echo the commands and their messages to the monitor. You can simply load your batch file into the editor or word processor you used to create it, then remove the *@echo off* command.

Another way to make sure you can see a batch file's commands is to use DOS' REM command. Just type *rem*, followed by a space, in front of the *@echo off* command like this:

```
rem @echo off
```

The REM notation tells DOS that the text that follows is a remark, instead of a command. Consequently, DOS won't execute the *@echo off* command.

After you've edited the batch file, save it, and then run it again. You'll be able to see DOS issue the command, and you'll also see where any error messages occur. Once you've debugged the batch file, simply remove the REM statement or restore *@echo off*, and DOS will once again suppress the display of commands.

Creating a disk cache with SMARTDrive can speed up your system

Somewhere on the front of your PC, you probably can see a small indicator light that flashes whenever your computer reads from or writes to its hard disk. The light is probably labelled HD (hard disk) or HDD (hard disk drive). When you're using DOS, you'll see it flash whenever you issue a command. For example, you'll see a very brief flash as it reads the files from the current directory when you type *dir* and press [Enter]. The light will stay on longer, however, when you copy a file from the hard disk to another drive. In fact, when you copy a very large file, you'll see the lights on your hard disk and diskette drives trade off a number of times as your computer copies the file in "manageable pieces." A manageable piece turns out to be the amount of information that DOS can load from your disk drive into random access memory (RAM). Once the information is in RAM, DOS can transfer it to the copy you're making.

You need RAM for a lot more than just copying files, however. In fact, your computer must place all pertinent information in RAM before it can process it. Once the information is in RAM, your computer can read or manipulate it very quickly—in many cases, almost instantaneously. Since RAM is so important—and so fast—you might wonder if you can store extra information there to speed up your system's performance. Fortunately, DOS 5 provides a disk-caching program called SMARTDrive, which can keep the data you most frequently need in RAM. In this article, we'll show you how to set up SMARTDrive 3.13 to create a cache on your PC. We'll also give you some background information on how disk caches can save you time and wear and tear on your hard disk.

Cache benefits

Programmers borrowed the term *cache* from an old word that means a storing place for provisions. By placing information it expects you'll need in RAM, a disk cache lives up to the old meaning of the word. Of course, a cache isn't intelligent enough to anticipate the information you'll need before you need it. Instead, a cache simply holds onto the information most recently read from the hard disk. This strategy is surprisingly effective in improving your system's performance.

When you use a cache, you'll notice the performance boost whenever you do something that repeatedly accesses the same information on the hard disk. For example, if you're searching for a name in a database, your PC will attempt to load into RAM either the entire database or an index to the database so that it can search through the entries to look for a match. If you're using a very large database, however, it won't fit into RAM. Instead, your application will have to read portions of the database from the disk into the available RAM, then keep track of all the records containing the name you're looking for.

In addition to database applications, you'll probably notice some improved speed in your spreadsheet and word processing applications. For example, depending on which program you use, your spelling checker may seem to run more quickly. Furthermore, a cache is so helpful to Windows performance that Microsoft includes SMARTDrive with Windows 3.0 and 3.1. (The sidebar "SMARTDrive and Windows" provides more information on the different SMARTDrive versions included in Windows.)

Ironically, you probably *won't* notice much of an improvement when you issue DOS commands while using SMARTDrive. Few DOS commands need to access repeatedly the same information from your hard disk. Still, chances are you'll notice improvement in your applications when you use SMARTDrive, so you'll probably want to install the driver, if your system supports it.

What you need to run SMARTDRV.SYS

You can create a SMARTDrive disk cache by installing the SMARTDRV.SYS driver through your CONFIG.SYS file. The information in the line varies, depending on whether you're creating the cache in extended or expanded memory and if you want to specify a particular size for the cache. But before you try to install SMARTDRV.SYS, you should make sure that your system will be able to use the driver.

Your system should have either extended or expanded memory to run SMARTDRV.SYS. Systems with less than 2 Mb of RAM can't spare the additional RAM that SMARTDRV reserves for the cache. If you're not familiar with extended and expanded memory, you might want to read the article "Understanding Your Computer's Memory," which begins on page 1, before you try to install SMARTDRV.SYS.

Even if you have enough memory to run SMARTDRV.SYS, you still may not be able to create the cache. You shouldn't try to use SMARTDRV.SYS if you also use a disk compression utility, such as the one provided with PC Tools. Also, SMARTDRV.SYS can't read some hard disks that make room for extra data by using special disk controllers. If you try to use SMARTDRV.SYS with one of these special hard disks, DOS will display the message *SMARTDrive: Too many bytes per sector on hard drive*. And, of course, you shouldn't run SMARTDRV.SYS if you're already using another disk-caching program.

The CONFIG.SYS file

As we said before, you install the SMARTDRV.SYS driver through your system's CONFIG.SYS file. (As you may know, the CONFIG.SYS file tells DOS how to configure your PC when it boots up.) But before it tells DOS that you want to use SMARTDRV.SYS, the CONFIG.SYS file must tell DOS to install the extended or expanded memory manager that you're using. Since the line that installs your memory manager should already be in your CONFIG.SYS file, you'll just need to make sure that you place the SMARTDRV.SYS line *after* the memory manager line. With that sequence in mind, let's look at the nuts and bolts of adding the SMARTDRV.SYS driver to your CONFIG.SYS file.

Your CONFIG.SYS file should be located in your PC's root directory. You can edit it using DOS 5's Editor, Edlin, or a word processor that allows you to save text as an ASCII-only file. If you have DOS 5, the Editor provides an easy way to edit the file. You can load CONFIG.SYS for editing by entering the command

```
C:\>edit config.sys
```

Then, you simply move the cursor below the line that installs your memory manager and type a DEVICE statement to install SMARTDRV.SYS. Since the line you use to install SMARTDRV.SYS varies depending on whether you're using extended or expanded memory, you need read only the section below that applies to the type of memory you'd use for the cache.

When you've finished editing the CONFIG.SYS file, save the changes you've made, then reboot. DOS will install SMARTDrive and create the disk cache.

Installing SMARTDRV.SYS in extended memory

Unless you tell it otherwise, DOS will assume that you want to install SMARTDRV.SYS in extended memory. By default, DOS will reserve 256 bytes of extended memory for the disk cache.

To create this cache, enter the following line in your CONFIG.SYS:

```
device=c:\dos\smartdrv.sys
```

When you add the SMARTDRV.SYS driver to your CONFIG.SYS file, be sure to place it after the line that installs your extended memory driver. Most DOS 5 users will use HIMEM.SYS, an extended memory manager included with the operating system. In that case, you

SMARTDrive and Windows

Since Microsoft now provides SMARTDrive with both Windows and DOS, those of you who have installed both products will need to decide which version of SMARTDrive to use. Windows 3.0 includes SMARTDrive 3.03, while DOS 5.0 comes with SMARTDrive 3.13, a minor upgrade to version 3.03. If you're using Windows 3.0, you can check your CONFIG.SYS file to see if Windows has installed SMARTDrive. Look for the statement

```
DEVICE=C:\WINDOWS\SMARTDRV.SYS
```

Since the SMARTDRV.SYS file is located in the Windows directory, you can assume that it is version 3.03. (Another way to check the version is to read the messages your CONFIG.SYS file generates as your PC boots up. If SMARTDRV.SYS is installed, you'll see a message that includes the SMARTDrive version number.)

If you want to install the more recent version of SMARTDrive that DOS 5 provides, simply edit your CONFIG.SYS file to specify the SMARTDRV.SYS found in the DOS directory. As you can see below, you do this by changing the C:\WINDOWS path to C:\DOS:

```
DEVICE=C:\DOS\SMARTDRV.SYS
```

Because *Inside DOS* focuses on DOS 5.0, the article "Creating a Disk Cache with SMARTDrive Can Speed Up Your System" explains how to use SMARTDRV.SYS 3.13, which DOS 5 provides. However, Windows 3.1 includes SMARTDrive 4.0, a major upgrade to the caching program. Instead of installing SMARTDRV.SYS through your CONFIG.SYS file, Windows 3.1 will run SMARTDRV.EXE through the AUTOEXEC.BAT file. If you've already installed SMARTDrive 4.0 by installing Windows 3.1, you don't need to install SMARTDrive through your CONFIG.SYS file, as described in the accompanying article. In fact, SMARTDrive 4.0 will include some features that SMARTDrive 3.13 doesn't have, such as a utility that reports the number of times it has found information from the cache.

could place the SMARTDRV.SYS driver after the HIMEM.SYS driver, as shown below:

```
device=c:\dos\himem.sys
device=c:\dos\smartdrv.sys
```

Of course, the SMARTDRV.SYS line doesn't have to come right after the HIMEM.SYS line—just be sure that HIMEM.SYS comes first.

Sizing the cache

As we said before, DOS will reserve 256 bytes by default for SMARTDRV.SYS when you create the disk cache in extended memory. You also can specify a larger disk cache, if you feel it's needed. (You could specify a smaller cache, but it's unlikely that you'd want to.) In general, you'd create a larger cache if you use applications that need to read data from the disk frequently. For example, you might want to specify a larger cache size if you work with large databases. Although the maximum cache size is 8192 bytes, Microsoft recommends that you choose a size between 256 bytes and 2048 bytes.

It's easy to resize the cache, so you can experiment with the size if you'd like. In fact, you'll probably notice a bigger improvement in your system's performance if you specify a cache size larger than the default of 256 bytes—especially if you work with very large files or databases. To choose a different size for the cache, you simply type a space, then the size, right after the *device=smartdrv.sys* statement. You type the size in bytes, not the more familiar kilobytes. (Remember, a kilobyte is 1024 bytes.) For example, to create a 2 Kb disk cache, you'd specify 2048 in the DEVICE statement:

```
device=c:\dos\smartdrv.sys 2048
```

If you happen to mistype the size of the cache, DOS will round off the value to the nearest multiple of 16. For example, if you typed 2038 instead of 2048, SMARTDRV.SYS would still create a cache of 2 Kb (2048 bytes).

Installing SMARTDRV.SYS in expanded memory

If your system has expanded memory, you can install SMARTDRV.SYS with a special switch to place the cache in expanded memory. When you place the command

```
device=c:\dos\smartdrv.sys /a
```

in your CONFIG.SYS file, DOS will create a cache that takes up all of your system's expanded memory. Remember to place this line after the line that installs your expanded memory device driver.

Although you can install the cache in the expanded memory simulated by EMM386.EXE, you probably won't notice as much improvement as you would if you installed the cache in "true" expanded memory from a special board installed in your computer, such as Intel Corp's AboveBoard. If you're using EMM386.EXE to simulate expanded memory and you still have some extended memory left in your system, you should probably install SMARTDRV.SYS in extended memory. In other words, leave off the /A switch. (If you're not sure about the type and amount of memory you have, we explain how to use the MEM command in the article "Understanding Your Computer's Memory," which begins on page 1.)

Sizing the cache

As we said before, DOS will assume that you want the cache to use all of your system's expanded memory when you install the driver with the statement

```
device=c:\dos\smartdrv.sys /a
```

However, if you have only expanded memory, you might not want the cache to reserve all of that memory. To prevent SMARTDRV.SYS from using all of your expanded memory, you can specify a size for the cache before the /A switch.

For example, suppose you have 2 Kb of expanded memory and you want to reserve only 1 Kb for the disk cache. You simply place 1024—the number of bytes in a kilobyte—between the driver name and the switch, as shown below:

```
device=c:\dos\smartdrv.sys 1024 /a
```

When it creates the cache, DOS will round off the size to the nearest multiple of 16. So, if you accidentally typed *1014*, DOS will create a disk cache of 1024 bytes.

Note

When you are sizing the disk cache, you also can specify a minimum cache size. This option is most useful when you run Windows. You can make sure that Windows won't reduce your cache size to zero by specifying a minimum size, such as 256 bytes. To do this, just type the minimum size after specifying a maximum size. For example, if you're using extended memory, you can specify a maximum size of 2048 bytes and a minimum size of 256 bytes by placing the following line in your CONFIG.SYS file:

```
device=c:\dos\smartdrv.sys 2048 256
```

To use expanded memory, you'd simply place the /A switch after minimum size.

Conclusion

A disk cache can help your system perform some tasks more quickly. In this article, we've shown you how to create a disk cache using SMARTDRV.SYS, a utility provided with DOS 5.

Continued from page 1

Understanding your computer's memory

Running programs

While RAM rarely performs tasks as simple as looking up a phone number, the analogy gives you an idea of how efficiently RAM works. In practice, one of the most important tasks RAM performs is holding the instructions for your programs. For example, when you run Microsoft Excel, DOS loads into RAM the instructions that make up the program EXCEL.EXE. When you issue commands in Excel, the program doesn't have to go back to the disk to look for EXCEL.EXE because the instructions to carry out the command are already in RAM.

Making changes

Although running program instructions is RAM's most important job, RAM also lets you temporarily store and manipulate data. For example, when you're editing a sentence in a word processing program, the program will load all or part of your document into RAM. If the application didn't load your document into RAM, you'd notice delays as you enter information because your computer would have to access its hard disk as you typed each letter.

Temporary storage

As we've seen, when you edit a document, you're really making changes to a copy of the document in RAM. You'll need to issue the program's save command to save your changes to the disk copy of the document. This brings us to RAM's Achilles' heel: Although it lets you run programs and quickly access and process data, it's not a permanent storage area. Unlike your system's hard disk, RAM "forgets" everything when you turn off your PC or reboot. That's why it's important to save your work frequently when you're entering or changing information. By saving the file, you write the new version to the hard disk, ensuring that your changes permanently become part of the file.

How much RAM do you have?

If you're not sure how much RAM, or "memory", is installed in your system, you can watch the self-test your computer goes through at boot up. (We describe the process in more detail in "How Your Computer Pulls Itself Up By Its Bootstraps," which appeared in the June 1992 issue of *Inside DOS*.) You should see a line that reports the base and extended memory installed in your PC. Almost every PC has one megabyte of memory, though it might report only 640 Kb when it boots up. The unreported 384 Kb is reserved as the *adapter segment*, a special part of memory DOS can use to control your system's hardware. For example, the driver for your monitor may be loaded into the adapter segment of RAM.

Although you can watch your system's memory check as it boots up, DOS 5 gives you a way of determining the amount of memory in your system at any time: the MEM command. To get a report on the memory in your system, you simply type

```
C:\>mem
```

and press [Enter]. DOS will present a report with several lines of information. When you want to find out how much memory you have in your system, you only need to pay attention to two of the lines. You simply add the number at the beginning of the first line (*bytes total conventional memory*) to the number at the beginning of the line *bytes total contiguous extended memory*. You'll probably come up with a number greater than one million. After you add the numbers together, you can divide the result by 1024 to convert it to the more familiar measure of kilobytes (Kb). Dividing again by 1024 will yield the amount of memory in megabytes (Mb).

For example, suppose the MEM command displayed this line for your system's conventional memory:

```
 655360  bytes total conventional memory
```

and this line for its contiguous extended memory:

```
3538944  bytes total contiguous extended memory
```

You add 655360 to 3538944 to find that your PC has 4194304 bytes of total memory. If you divide that result by 1024, you'll find that you have 4096 Kb of RAM. Or, if you'd prefer the result in megabytes, you can divide by 1024 again to find out that you have a total of 4 Mb of RAM.

Where does RAM come from?

When you buy a computer, you usually have to pay extra for memory beyond the first megabyte. That first, "standard" megabyte is often called *conventional memory*. Some companies, however, sell configurations with additional memory as "packages," so you might not notice the breakdown in price.

You also can add memory to a computer after you've purchased it. Usually, PC users decide to add memory after they've upgraded to a program that puts more demands on memory, such as Microsoft Windows or a desktop publishing program. Although some models of computers allow you to add memory simply by plugging chips into additional sockets, many models will require you—or a technician—to install a memory board. If you're thinking of upgrading your PC, you should contact the company that sold you the computer or the manufacturer to find out exactly what you need and how much it will cost to add memory.

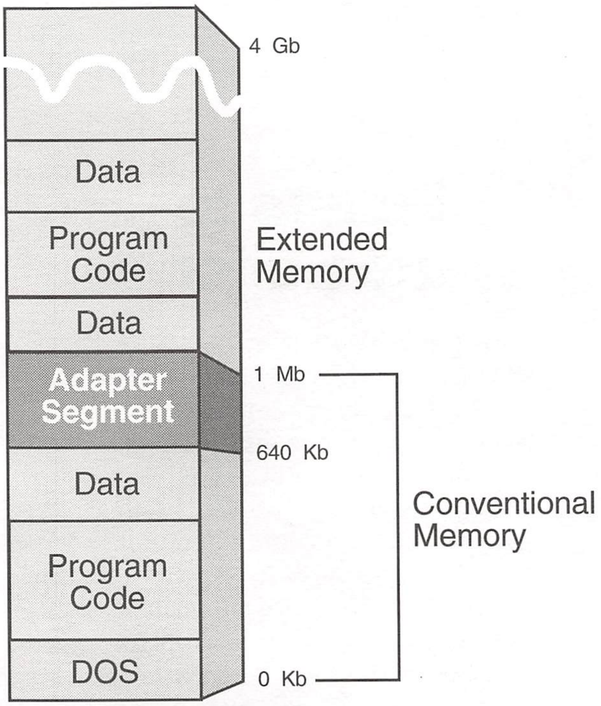
What's the difference between extended and expanded memory?

Memory boards can either provide extended or expanded memory—or both. To make matters more confusing, you can use software to make your computer use extended memory as expanded memory. Both extended memory and expanded memory describe memory beyond the first megabyte installed in your PC. The difference between the two types is how your applications can access the memory.

Extended memory

You can think of extended memory as memory that is stacked above the first megabyte, as shown in Figure A. While 80286, 80386, and 80486 central processing units ("chips") have the potential to access this extended memory, DOS can't access the memory on its own. Some applications, such as Windows, get around DOS' limitation and use extended memory directly.

Figure A



Extended memory lies above the first megabyte of RAM.

Fortunately, DOS 5 includes a special driver, HIMEM.SYS, which allows DOS to use your system's extended memory. You install the HIMEM.SYS driver by placing the following line in your CONFIG.SYS file:

```
device=c:\dos\himem.sys
```

In general, we recommend that you place this line at the top of your CONFIG.SYS file. After carrying out the HIMEM.SYS instruction, your computer will be able to use extended memory for other CONFIG.SYS statements, if necessary. (Also, note that we use a DEVICE statement, rather than a DEVICEHIGH statement.)

After you've saved the new version of your CONFIG.SYS file, reboot. DOS will install HIMEM.SYS so that it can access your system's extended memory. Now that you've installed HIMEM.SYS, your memory meets the eXtended Memory Specification, or XMS. You might need to know that you have XMS when you install new software.

Expanded memory

Even after you've installed HIMEM.SYS so that DOS can access extended memory, some of your applications still might not be able to use the additional memory. This is because some applications (particularly older ones) can't use extended memory at all. Instead, these applications need *expanded* memory, which meets the Lotus-Intel-Microsoft Expanded Memory Specification (LIM EMS).

Even though it's more "primitive" in a way, expanded memory seems a bit more complex than extended memory. Instead of simply stacking additional memory on top of the first megabyte of RAM, expanded memory divides the additional memory into segments, each containing 16 Kb of memory. Earlier, we mentioned that DOS reserves part of the first megabyte of RAM as the adapter segment, which contains instructions to help DOS manage hardware. Expanded memory uses part of the adapter segment to read a "page" of data into your system's conventional memory, as you can see in Figure B on page 12. This effect is similar to "paging" through a large document in your word processing program. Each time you press [PageDown], you bring another screen of data into view.

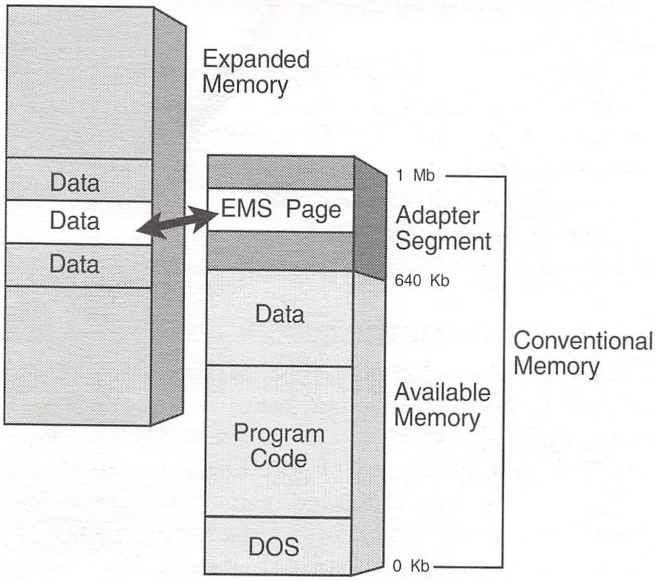
Suppose you have extended memory that you manage with HIMEM.SYS, but you run an application that requires expanded memory. Fortunately, Microsoft now includes an expanded memory manager, EMM386.EXE, with DOS 5. This program uses part of extended memory to simulate the page-swapping of expanded memory. You install the EMM386.EXE driver by adding the following line to your CONFIG.SYS file:

```
device=c:\dos\emm386.exe
```

Microsoft Technical Support
(206) 454-2030

Please include account number from label with any correspondence.

Figure B



The EMS memory manager swaps pages of information from expanded memory into and out of conventional memory.

Since EMM386.EXE uses extended memory to simulate expanded memory, be sure to place this line *after* the line that installs HIMEM.SYS. When you reboot, DOS will convert 256 Kb of your system's extended memory to expanded memory. If you issue the MEM command, you'll see two lines reporting on the expanded memory (alias EMS) that you've installed:

```
655360 bytes total EMS memory
262144 bytes free EMS memory
```

As you can see, the MEM command reports 640 Kb of conventional memory (or 655360 bytes) as the total EMS memory. The figure for free EMS memory is what EMM386.EXE actually sets aside for expanded memory—256 Kb, or 262144 bytes.

If you have trouble running some applications after you've installed EMM386.EXE, consult the application's documentation to see if they recommend that you specify any particular values when you install EMM386.EXE. For example, a database program that uses only expanded memory may need more than the 256 Kb default that EMM386.EXE provides. You can create 1 Mb (that is, 1024 Kb) of expanded memory by adding *1024* to the EMM386.EXE statement

```
device=c:\dos\emm386.exe 1024
```

When you reboot, DOS will convert 1 Mb of extended memory into expanded memory for your applications to use.

Conclusion

Random access memory is one of the most important elements in your computer. RAM holds the instructions for the programs you run, and it also holds data so that you can process it. In this article, we have answered some basic questions about RAM. We've also compared extended and expanded memory and shown you the basics of managing memory with the device drivers that DOS 5 provides. In a future article, we'll look at other aspects of memory, including how you load devices into high memory to free up more conventional memory.

CUSTOMER FYI RELATIONS

Letters to the Editor

We value your questions and article idea submissions. Please include samples of documents that illustrate any problem or idea that you want to share. We'll publish as many letters as our journals can accommodate. If many people request information on one topic, we'll cover it in an article.

If you'd like to receive information about the types of articles we accept for publication, please contact our Customer Relations personnel, who will be happy to help you. Call toll free (800) 223-8720 or (502) 491-1900 or fax us at (502) 491-8050.

Printed on Halopaque: 100% acid free, made from 50% de-inked, recycled fiber including 10% post-consumer waste. This journal is 100% recyclable.